Prediction strategies without loss

Michael Kapralov
Stanford University
Stanford, CA
kapralov@stanford.edu

Rina Panigrahy
Microsoft Research Silicon Valley
Mountain View, CA
rina@microsoft.com

Abstract

Consider a sequence of bits where we are trying to predict the next bit from the previous bits. Assume we are allowed to say 'predict 0' or 'predict 1', and our payoff is $+1$ if the prediction is correct and -1 otherwise. We will say that at each point in time the loss of an algorithm is the number of wrong predictions minus the number of right predictions so far. In this paper we are interested in algorithms that have essentially zero (expected) loss over any string at any point in time and yet have small regret with respect to always predicting 0 or always predicting 1. For a sequence of length T our algorithm has regret $14\epsilon T$ and loss $2\sqrt{T}e^{-\epsilon^2 T}$ in expectation for all strings. We show that the tradeoff between loss and regret is optimal up to constant factors.

Our techniques extend to the general setting of N experts, where the related problem of trading off regret to the best expert for regret to the 'special' expert has been studied by Even-Dar et al. (COLT'07). We obtain essentially zero loss with respect to the special expert and optimal loss/regret tradeoff, improving upon the results of Even-Dar et al and settling the main question left open in their paper.

The strong loss bounds of the algorithm have some surprising consequences. First, we obtain a parameter free algorithm for the experts problem that has optimal regret bounds with respect to k-shifting optima, i.e. bounds with respect to the optimum that is allowed to change arms multiple times. Moreover, for *any window of size* n the regret of our algorithm to any expert never exceeds $O(\sqrt{n(\log N + \log T)})$, where N is the number of experts and T is the time horizon, while maintaining the essentially zero loss property.

1 Introduction

Consider a gambler who is trying to predict the next bit in a sequence of bits. One could think of the bits as indications of whether a stock price goes up or down on a given day, where we assume that the stock always goes up or down by 1 (this is, of course, a very simplified model of the stock market). If the gambler predicts 1 (i.e. that the stock will go up), she buys one stock to sell it the next day, and short sells one stock if her prediction is 0. We will also allow the gambler to bet fractionally by letting him specify a confidence c where $0 \leq c \leq 1$ in his prediction. If the prediction is right the gambler gets a payoff of c otherwise $-c$. While the gambler is tempted to make predictions with the prospect of making money, there is also the risk of ending up with a loss. Is there a way to never end up with a loss? Clearly there is the strategy of never predicting (by setting confidence 0) all the time that never has a loss but also never has a positive payoff. However, if the sequence is very imbalanced and has many more 0's than 1's then this never predict strategy has a high regret with respect to the strategy that predicts the majority bit. Thus, one is interested in a strategy that has a small regret with respect to predicting the majority bit and incurs no loss at the same time.

Our main result is that while one cannot always avoid a loss and still have a small regret, this is possible if we allow for an *exponentially small* loss. More precisely, we show that for any $\epsilon > 1/\sqrt{T}$

there exists an algorithm that achieves regret at most $14\epsilon T$ and loss at most $2e^{-\epsilon^2 T}\sqrt{T}$, where T is the time horizon. Thus, the loss is *exponentially small in the length of the sequence*.

The bit prediction problem can be cast as the experts problem with two experts: S_+, that always predicts 1 and S_- that always predicts 0. This problem has been studied extensively, and very efficient algorithms are known. The weighted majority algorithm of [12] is known to give optimal regret guarantees. However, it can be seen that weighted majority may result in a loss of $\Omega(\sqrt{T})$. The best known result on bounding loss is the work of Even-Dar et al. [7] on the problem of trading off regret to the best expert for regret to the average expert, which is equivalent to our problem. Stated as a result on bounding loss, they were able to obtain a constant loss and regret $O(\sqrt{T \log T})$. Their work left the question open as to whether it is possible to even get a regret of $O(\sqrt{T \log T})$ and constant loss. In this paper we give an optimal regret/loss tradeoff, in particular showing that this regret can be achieved even with subconstant loss.

Our results extend to the general setting of prediction with expert advice when there are multiple experts. In this problem the decision maker iteratively chooses among N available alternatives without knowledge of their payoffs, and gets payoff based on the chosen alternative. The payoffs of all alternatives are revealed after the decision is made. This process is repeated over T rounds, and the goal of the decision maker is to maximize her cumulative payoff over all time steps $t = 1, \ldots, T$. This problem and its variations has been studied extensively, and efficient algorithms have been obtained (e.g. [5, 12, 6, 2, 1]). The most widely used measure of performance of an online decision making algorithm is *regret*, which is defined as the difference between the payoff of the best *fixed* alternative and the payoff of the algorithm. The well-known weighted majority algorithm of [12] obtains regret $O(\sqrt{T \log N})$ even when no assumptions are made on the process generating the payoff. Regret to the best fixed alternative in hindsight is a very natural notion when the payoffs are sampled from an unknown distribution, and in fact such scenarios show that the bound of $O(\sqrt{T \log N})$ on regret achieved by the weighted majority algorithm is optimal.

Even-Dar et al. [7] gave an algorithm that has *constant* regret to any fixed distribution on the experts at the expense of regret $O(\sqrt{T \log N}(\log T + \log\log N))$ with respect to all other experts[1]. We obtain an optimal tradeoff between the two, getting an algorithm with regret $O(\sqrt{T(\log N + \log T)})$ to the best and $O((NT)^{-\Omega(1)})$ to the average as a special case. We also note, similarly to [7] that our regret/loss tradeoff cannot be obtained by using standard regret minimization algorithms with a prior that is concentrated on the 'special' expert, since the prior would have to put a significant weight on the 'special' expert, resulting in $\Omega(T)$ regret to the best expert.

The extension to the case of N experts uses the idea of improving one expert's predictions by that of another. The strong loss bounds of our algorithm allow us to achieve *lossless boosting*, i.e. we use available expert to continuously improve upon the performance of the base expert whenever possible while essentially never hurting its performance. When comparing two experts, we track the difference in the payoffs discounted geometrically over time and apply a transform $g(x)$ on this difference to obtain a weighting that is applied to give a linear combination of the two experts with a higher weight being applied on the expert with a higher discounted payoff. The shape of $g(x)$ is given by $\mathrm{erf}\left(\frac{x}{4\sqrt{T}}\right) e^{x^2/(16T)}$, capped at ± 1. The weighted majority algorithm on the other hand uses a transform with the shape of the $\tanh(\frac{x}{\sqrt{T}})$ function and ignores geometric discounting (see Figure 1).

An important property of our algorithm is that it does not need a high imbalance between the number of ones and the number of zeros in the whole sequence to have a gain: it is sufficient for the imbalance to be large enough in *at least one contiguous time window*[2], the size of which is a parameter of the algorithm. This property allows us to easily obtain optimal *adaptive regret* bounds, i.e. we show that the payoff of our algorithm in *any geometric window* of size n is at most $O(\sqrt{n \log(NT)})$

[1] In fact, [7] provide several algorithms, of which the most relevant for comparison are *Phased Agression*, yielding $O(\sqrt{T \log N}(\log T + \log\log N))$ regret to the best and D-Prod, yielding $O(\sqrt{T/\log N}\log T)$ regret to the best. For the bit prediction problem one would set $N = 2$ and use the uniform distribution over the 'predict 0' and 'predict 1' strategy as the special distribution. Our algorithm improves on both of them, yielding an optimal tradeoff.

[2] More precisely, we use an infinite window with geometrically decreasing weighting, so that most of the weight is contained in the window of size $O(n)$, where n is a parameter of the algorithm.

worse than the payoff of the strategy that is *best in that window* (see Theorem 11). In the full version of the paper ([11]) we also obtain bounds against the class of strategies that are allowed to change experts multiple times while maintaining the essentially zero loss property. We note that even though similar bounds (without the essentially zero loss property) have been obtained before ([3, 9, 14] and, more recently, [10]), our approach is very different and arguably simpler.

In the full version of the paper, we also show how our algorithm yields regret bounds that depend on the l_p norm of the costs, regret bounds dependent on Kolmogorov complexity as well as applications of our framework to multi-armed bandits with partial information and online convex optimization.

1.1 Related work

The question of what can be achieved if one would like to have a significantly better guarantee with respect to a fixed arm or a distribution on arms was asked before in [7] as we discussed in the introduction. Tradeoffs between regret and loss were also examined in [13], where the author studied the set of values of a, b for which an algorithm can have payoff $aOPT + b \log N$, where OPT is the payoff of the best arm and a, b are constants. The problem of bit prediction was also considered in [8], where several loss functions are considered. None of them, however, corresponds to our setting, making the results incomparable.

In recent work on the NormalHedge algorithm[4] the authors use a potential function which is very similar to our function $g(x)$ (see (2) below), getting strong regret guarantees to the ϵ-quantile of best experts. However, the use of the function $g(x)$ seems to be quite different from ours, as is the focus of the paper [4].

1.2 Preliminaries

We start by defining the bit prediction problem formally. Let $b_t, t = 1, \ldots, T$ be an adversarial sequence of bits. It will be convenient to adopt the convention that $b_t \in \{-1, +1\}$ instead of $b_t \in \{0, 1\}$ since it simplifies the formula for the payoff. In fact, in what follows we will only assume that $-1 \leq b_t \leq 1$, allowing b_t to be real numbers. At each time step $t = 1, \ldots, T$ the algorithm is required to output a *confidence level* $f_t \in [-1, 1]$, and then the value of b_t is revealed to it. The payoff of the algorithm by time t' is

$$A_{t'} = \sum_{t=1}^{t'} f_t b_t. \tag{1}$$

For example, if $b_t \in \{-1, +1\}$, then this setup is analogous to a prediction process in which a player observes a sequence of bits and at each point in time predicts that the value of the next bit will be $\text{sign}(f_t)$ with confidence $|f_t|$. Predicting $f_t \equiv 0$ amounts to not playing the game, and incurs no loss, while not bringing any profit. We define the loss of the algorithm on a string b as

$$\text{loss} = \min\{-A_t, 0\},$$

i.e. the absolute value of the smallest negative payoff over all time steps.

It is easy to see that any algorithm that has a positive expected payoff on some sequence necessarily loses on another sequence. Thus, we are concerned with finding a prediction strategy that has *exponentially small loss bounds* but also has *low regret* against a number of given prediction strategies. In the simplest setting we would like to design an algorithm that has low regret against two basic strategies: S_+, which always predicts $+1$ and S_-, which always predicts -1. Note that the maximum of the payoffs of S_+ and S_- is always equal to $\left|\sum_{t=1}^{T} b_t\right|$. We denote the base random strategy, which predicts with confidence 0, by S_0. In what follows we will use the notation A_T for the cumulative payoff of the algorithm by time T as defined above.

As we will show in section 3, our techniques extend easily to give an algorithm that has low regret with respect to the best of any N bit prediction strategies and exponentially small loss. Our techniques work for the general experts problem, where loss corresponds to regret with respect to the 'special' expert S_0, and hence we give the proof in this setting. This provides the connection to the work of [7].

In section 2 we give an algorithm for the case of two prediction strategies S_+ and S_-, and in section 3 we extend it to the general experts problem, additionally giving the claimed adaptive regret bounds.

2 Main algorithm

The main result of this section is

Theorem 1 *For any $\epsilon \geq \sqrt{\frac{1}{T}}$ there exists an algorithm A for which*

$$A_T \geq \max\left\{\left|\sum_{j=1}^{T} b_j\right| - 14\epsilon T, 0\right\} - 2\sqrt{T}e^{-\epsilon^2 T},$$

i.e. the algorithm has at most $14\epsilon T$ regret against S_+ and S_- as well as a exponentially small loss. *By setting ϵ so that the loss bound is $2Z\sqrt{T}$, we get a regret bound of $\sqrt{T\log(1/Z)}$.*

We note that the algorithm is a strict generalization of weighted majority, which can be seen by letting $Z = \Theta(1)$ (this property will also hold for the generalization to N experts in section 3).

Our algorithm will have the following form. For a chosen discount factor $\rho = 1 - 1/n, 0 \leq \rho \leq 1$ the algorithm maintains a discounted deviation $x_t = \sum_{j=1}^{t-1} \rho^{t-1-j} b_j$ at each time $t = 1, \ldots, T$. The value of the prediction at time t is then given by $g(x_t)$ for a function $g(\cdot)$ to be defined (note that x_t depends only on $b_{t'}$ for $t' < t$, so this is an online algorithm). The function g as well as the discount factor ρ depend on the desired bound on expected loss and regret against S_+ and S_-. In particular, we will set $\rho = 1 - \frac{1}{T}$ for our main result on regret/loss tradeoff, and will use the freedom to choose different values of ρ to obtain adaptive regret guarantees in section 3. The algorithm is given by

Algorithm 1: Bounded loss prediction

1: $x_1 \leftarrow 0$
2: **for** $t = 1$ to T **do**
3: Predict $\text{sign}(g(x_t))$ with confidence $|g(x_t)|$.
4: Set $x_{t+1} \leftarrow \rho x_t + b_t$.
5: **end for**

We start with an informal sketch of the proof, which will be made precise in Lemma 2 and Lemma 3 below. The proof is based on a potential function argument. In particular, we will choose the confidence function $g(x)$ so that

$$\Phi_t = \int_0^{x_t} g(s)ds.$$

is a potential function, which serves as a repository for guarding our loss (we will chose $g(x)$ to be an odd function, and hence will always have $\Phi_t \geq 0$). In particular, we will choose $g(x)$ so that the change of Φ_t lower bounds the payoff of the algorithm. If we let $\Phi_t = G(x_t)$ (assuming for sake of clarity that $x_t > 0$), where

$$G(x) = \int_0^{x} g(s)ds,$$

we have

$$\Phi_{t+1} - \Phi_t = G(x_{t+1}) - G(x_t) \approx G'(x)\Delta x + G''(x)\Delta x^2/2 \approx g(x)\left[(\rho - 1)x + b_t\right] + g'(x)/2.$$

Since the payoff of the algorithm at time step t is $g(x_t)b_t$, we have

$$\Delta\Phi_t - g(x_t)b_t = -g(x_t)(1-\rho)x_t + g'(x_t)/2,$$

so the condition becomes

$$-g(x_t)(1-\rho)x_t + g'(x_t)/2 \leq Z,$$

where Z is the desired bound on per step loss of the algorithm. Solving this equation yields a function of the form

$$g(x) = (2Z\sqrt{T}) \cdot \text{erf}\left(\frac{x}{4\sqrt{T}}\right) e^{x^2/(16T)},$$

where $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-s^2} ds$ is the error function (see Figure 1 for the shape of $g(x)$).

We now make this proof sketch precise. For $t = 1, \ldots, T$ define $\Phi_t = \int_0^{x_t} g(x)dx$. The function $g(x)$ will be chosen to be a continuous odd function that is equal to 1 for $x > U$ and to -1 when $x < -U$, for some $0 < U < T$. Thus, we will have that $|x_t| - U \leq \Phi_t \leq |x_t|$. Intuitively, Φ_t captures the imbalance between the number of -1's and $+1$'s in the sequence up to time t.

We will use the following parameters. We always have $\rho = 1 - 1/n$ for some $n > 1$ and use the notation $\bar{\rho} = 1 - \rho$. We will later choose $n = T$ to prove Theorem 1, but we will use different value of n for the adaptive regret guarantees in section 3.

We now prove that if the function $g(x)$ approximately satisfies a certain differential equation, then Φ_t defined as above is a potential function. The statement of Lemma 2 involves a function $h(x)$ that will be chosen as a step function that is 1 when $x \in [-U, U]$ and 0 otherwise.

Lemma 2 *Suppose that the function $g(x)$ used in Algorithm 1 satisfies*

$$\frac{1}{2}(\bar{\rho}x + 1)^2 \cdot \max_{s \in [\rho x - 1, \rho x + 1]} |g'(s)| \leq \bar{\rho} x g(x) h(x) + Z'$$

for a function $h(x), 0 \leq h(x) \leq 1, \forall x$, for some $Z' > 0$. Then the payoff of the algorithm is at least

$$\sum_{t=1}^{T} \bar{\rho} x_t g(x_t)(1 - h(x)) + \Phi_{T+1} - Z'T$$

as long as $|b_t| \leq 1$ for all t.

Proof: We will show that at each t

$$\Phi_{t+1} - \Phi_t \leq b_t g(x_t) + Z' - \bar{\rho} x_t g(x_t)(1 - h(x_t)),$$

i.e.

$$\sum_{t=1}^{T} b_t g(x_t) \geq -Z'T + \sum_{t=1}^{T} \bar{\rho} x_t g(x_t)(1 - h(x_t)) + \Phi_{T+1} - \Phi_1,$$

thus implying the claim of the lemma since $\Phi_1 = 0$.

We consider the case $x_t > 0$. The case $x_t < 0$ is analogous. In the following derivation we will write $[A, B]$ to denote $[\min\{A, B\}, \max\{A, B\}]$.

$0 \leq b_t \leq 1$**:** We have $x_{t+1} = \rho x_t + b_t = x_t - \bar{\rho} x_t + b_t$, and the expected payoff of the algorithm is $g(x_t) b_t$. Then

$$\begin{aligned}
\Phi_{t+1} - \Phi_t &= \int_{x_t}^{x_t - \bar{\rho} x_t + b_t} g(s) ds \\
&\leq g(x_t)(b_t - \bar{\rho} x_t) + \frac{1}{2}(\bar{\rho} x_t + b_t)^2 \cdot \max_{s \in [x_t, x_t - \bar{\rho} x_t + b_t]} |g'(s)| \\
&\leq g(x_t) b_t + \left[-g(x_t) \bar{\rho} x_t + \frac{1}{2}(\bar{\rho} x_t + 1)^2 \cdot \max_{s \in [x_t, x_t - \bar{\rho} x_t + b_t]} |g'(s)| \right] \\
&\leq g(x_t) b_t + (-1 + h(x_t)) \bar{\rho} x_t g(x_t) + Z'.
\end{aligned}$$

$-1 \leq b_t \leq 0$**:** This case is analogous. ■

We now define $g(x)$ to satisfy the requirement of Lemma 2. For any $Z, L > 0$ and let

$$g(x) = \text{sign}(x_t) \cdot \min \left\{ Z \cdot \text{erf}\left(\frac{|x|}{4L} \right) e^{\frac{x^2}{16L^2}}, 1 \right\}. \tag{2}$$

One can show that one has $g(x) = 1$ for $|x| \geq U$ for some $U \leq 7L\sqrt{\log(1/Z)}$. A plot of the function $g(x)$ is given in Figure 1.

We choose

$$h(x) = \begin{cases} 1, & |x| < U \\ 0 & \text{o.w.} \end{cases}. \tag{3}$$

The following lemma shows that the function $g(x)$ satisfies the properties stated in Lemma 2:

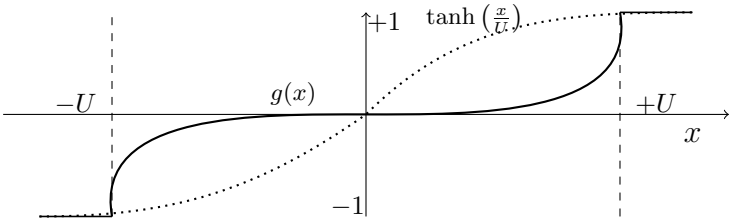

Figure 1: The shape of the confidence function $g(x)$ (solid line) and the $\tanh(x)$ function used by weighted majority (dotted line).

Lemma 3 *Let $L > 0$ be such that $\bar{\rho} = 1/n \geq 1/L^2$. Then for $n \geq 80 \log(1/Z)$ the function $g(x)$ defined by* (2) *satisfies*

$$\frac{1}{2}(\bar{\rho}x + 1)^2 \cdot \max_{s \in [\rho x - 1, \rho x + 1]} |g'(s)| \leq \bar{\rho} x g(x) h(x)/2 + 2\bar{\rho} L Z,$$

where $h(x)$ is the step function defined above.

Proof: The intuition behind the Lemma is very simple. Note that $s \in [\rho x - 1, \rho x + 1]$ is not much further than 1 away from x, so $g'(s)$ is very close to $g'(x) = (\frac{x}{2L^2})g(x) + \frac{1}{\sqrt{\pi}L}Z$. Since $\bar{\rho} \geq 1/L^2$, we have $g'(x) \leq \bar{\rho} x g(x)/2 + \frac{1}{\sqrt{\pi}}\bar{\rho} L Z$. We defer the details of the proof to the full version. ■

We can now lower bound the payoff of Algorithm 1. We will use the notation

$$|x|_{\epsilon}^{+} = \begin{cases} 0, & |x| < \epsilon \\ |x| - \epsilon, & |x| > \epsilon \end{cases}$$

Theorem 4 *Let n be the window size parameter of Algorithm 1. Then one has*

$$A_T \geq \sum_{t=1}^{T} \bar{\rho} |x_t|_U^+ + |x_{T+1}|_U^+ - 2ZT/\sqrt{n}.$$

Proof: By Lemma 3 we have that the function $g(x)$ satisfies the conditions of Lemma 2, and so from the bounds stated in Lemma 2 the payoff of the algorithm is at least

$$\sum_{t=1}^{T} \bar{\rho} |x_t|_U^+ + \Phi_{T+1} - 2ZT/\sqrt{n}.$$

By definition of Φ_t, since $|g(x)| = 1$ for $|x| \geq U$, one has $\Phi_{T+1} \geq |x_{T+1}|_U^+$, which gives the desired statement. ■

Now, setting $n = T$, we obtain

Theorem 5

$$A_T \geq \max\left\{ \left| \sum_{j=1}^{T} b_j \right| - 14\sqrt{T \log(1/Z)}, 0 \right\} - 2Z\sqrt{T}.$$

Proof: In light of Theorem 4 it remains to bound $\sum_{t=1}^{T} \bar{\rho} x_t + x_{T+1}$. We have

$$\bar{\rho} \sum_{t=1}^{T} x_t + x_{T+1} = \bar{\rho} \sum_{t=1}^{T-1} \sum_{j=1}^{t} \rho^{t-j} b_j + x_{T+1} = \sum_{t=1}^{T-1} b_t (1 - \rho^{T-t}) + \sum_{t=1}^{T} \rho^{T-t} b_t = \sum_{t=1}^{T} b_t. \quad (4)$$

Thus, since $U \leq 2\sqrt{T \log(1/Z)}$, and we chose $\rho = 1 - 1/n = 1 - 1/T$, we get the result by combining Theorem 4 and equation (4). ■

Proof of Theorem 1: Follows by setting $\log(1/Z) = \epsilon^2 T$. ■

Our loss/regret tradeoff is optimal up to constant factors (proof deferred to the full version):

Theorem 6 *Any algorithm that has regret $O(\sqrt{T\log(1/Z)})$ incurs loss $\Omega(Z\sqrt{T})$ on at least one sequence of bits $b_t, t = 1, \ldots, T$.*

Note that if $Z = o(1/T)$, then the payoff of the algorithm is positive whenever the absolute value of the deviation x_t is larger than, say $8\sqrt{n\log T}$ *in at least one window of size* n.

3 Combining strategies (lossless boosting)

In the previous section we derived an algorithm for the bit prediction problem with low regret to the S_+ and S_- strategies and exponentially small loss. We now show how our techniques yield an algorithm that has low regret to the best of N bit prediction strategies $S_1, \ldots, S_N$ and exponentially small loss. However, since the proof works for the general experts problem, where loss corresponds to regret to a 'special' expert S_0, we state it in the general experts setting. In what follows we will refer to regret to S_0 as loss. We will also prove optimal bounds on regret that hold *in every window of length* n at the end of the section. We start by proving

Theorem 7 *For any $Z < 1/e$ there exists an algorithm for combining N strategies that has regret $O(\sqrt{T\log(N/Z)})$ against the best of N strategies and loss at most $O(ZN\sqrt{T})$ with respect to any strategy S_0 fixed a priori. These bounds are optimal up to constant factors.*

We first fix notation. A *prediction strategy* S given a bit string b_t, produces a sequence of weights w_{jt} on the set of experts $j = 1, \ldots, N$ such that w_{jt} depends only on $b_{t'}, t' < t$ and $\sum_{j=1}^{N} w_{jt} = 1, w_{jt} \geq 0$ for all t. Thus, using strategy S amounts to using expert j with probability $w_{j,t}$ at time t, for all $t = 1, \ldots, T$. For two strategies S_1, S_2 we write $\alpha_t S_1 + (1-\alpha_t)S_2$ to denote the strategy whose weights are a convex combination of weights of S_1 and S_2 given by coefficients $\alpha_t \in [0, 1]$. For a strategy S we denote its payoff at time t by s_t.

We start with the case of two strategies S_1, S_2. Our algorithm will consider S_1 as the base strategy (corresponding to the null strategy S_0 in the previous section) and will use S_2 to improve on S_1 whenever possible, without introducing significant loss over S_1 in the process. We define

$$\bar{g}(x) = \begin{cases} g(\frac{1}{2}x), & x > 0 \\ 0 & \text{o.w,} \end{cases}$$

i.e. we are using a one-sided version of $g(x)$. It is easy to see that $\bar{g}(x)$ satisfies the conditions of Lemma 2 with $h(x)$ as defined in (3). The intuition behind the algorithm is that since the difference in payoff obtained by using S_2 instead of S_1 is given by $(s_{2,t} - s_{1,t})$, it is sufficient to emulate Algorithm 1 on this sequence. In particular, we set $x_t = \sum_{j=1}^{t-1} \rho^{t-1-j}(s_{2,j} - s_{1,j})$ and predict $\bar{g}(x_t)$ (note that since $|s_{1,t} - s_{2,t}| \leq 2$, we need to use $g(\frac{1}{2}x)$ in the definition of $\bar{g}$ to scale the payoffs). Predicting 0 corresponds to using S_1, predicting 1 corresponds to using S_2 and fractional values correspond to a convex combination of S_1 and S_2.

Formally, the algorithm COMBINE(S_1, S_2, ρ) takes the following form:

Algorithm 2: COMBINE(S_1, S_2, ρ)

```
Input: strategies S_1, S_2
Output: strategy S*
x_1 ← 0
for t = 1 to T do
    Set S*_t ← S_{1,t}(1 − ḡ(x_t)) + S_{2,t} ḡ(x_t).
    Set x_{t+1} ← ρ x_t + (s_{2,t} − s_{1,t}).
end for
return S*
```

Note that COMBINE(S_1, S_2, ρ) is an online algorithm, since S_t^* only depends on $s_{1,t'}, s_{2,t'}, t' < t$.

Lemma 8 *There exists an algorithm that given two strategies S_1 and S_2 gets payoff at least*

$$\sum_{t=1}^{T} s_{1,t} + \max\left\{\sum_{t=1}^{T}(s_{2,t}-s_{1,t}) - O\left(\sqrt{T\log(1/Z)}\right), 0\right\} - O(Z\sqrt{T}).$$

PROOF OUTLINE: Use Algorithm 2 with $\rho = 1 - 1/T$. This amounts to applying Algorithm 1 to the sequence $(s_{2,t} - s_{1,t})$, so the guarantees follow by Theorem 4. ■

We emphasize the property that Algorithm 2 combines two strategies S_1 and S_2, improving on the performance of S_1 using S_2 whenever possible, *essentially without introducing any loss with respect to* S_1. Thus, this amounts to *lossless boosting* of one strategy's performance using another. Thus, we have

Proof of Theorem 7: Use Algorithm 2 repeatedly to combine N strategies $S_1, \ldots, S_N$ by initializing $S^0 \leftarrow S_0$ and setting $S^j \leftarrow \text{COMBINE}(S^{j-1}, S_j, 1-1/T), j = 1, \ldots, N$, where S_0 is the null strategy. The regret and loss guarantees follow by Lemma 8. ■

Corollary 9 *Setting $Z = (NT)^{-1-\gamma}$ for $\gamma > 0$, we get regret $O(\sqrt{\gamma T(\log N + \log T)})$ to the best of N strategies and loss at most $O((NT)^{-\gamma})$ wrt strategy S_0 fixed a priori. These bounds are optimal and improve on the work on [7].*

So far we have used $\rho = 1 - 1/T$ for all results. One can obtain optimal adaptive guarantees by performing boosting over a range of decay parameters ρ. In particular, choose $\rho_j = 1 - n_j$, where $n_j, j = 1, \ldots, W$ are powers of two between $80\log(NT)$ and T. Then let

Algorithm 3: Boosting over different time scales

```
S^{0,W} ← S_0
for j = W downto 1 do
    for k = 1 to N do
        S^{k,j} ← COMBINE(S^{k-1,j}, S_k, 1 − 1/n_j)
    end for
    S^{0,j-1} ← S^{N,j}
end for
S* ← S^{0,0}
return S*
```

We note that it is important that the outer loop in Algorithm 3 goes from large windows down to small windows. Finally, we show another adaptive regret property of Algorithm 3. First, for a sequence $w_t, t = 1, \ldots, T$ of real numbers and for a parameter $\rho = 1 - 1/n \in (0,1)$ define

$$\tilde{w}_t^{\rho} = \sum_{j=1}^{t} \rho^{t-j} w_j.$$

We will need the following definition:

Definition 10 *A sequence w_j is Z-*uniform at scale $\rho = 1 - 1/n$ *if one has $\widetilde{w_t}^{\rho} \le c\sqrt{n\log(1/Z)}$ for all $1 \le t \le T$, for some constant $c > 0$.*

Note that if the input sequence is iid $\textbf{Ber}(\pm 1, 1/2)$, then it is Z-uniform at any scale with probability at least $1 - Z$ for any $Z > 0$. We now prove that the difference between the payoff of our algorithm and the payoff of any expert is Z-uniform, i.e. does not exceed the standard deviation of a uniformly random variable in any sufficiently large window, when the loss is bounded by Z. More precisely,

Theorem 11 *The sequences $s_{j,t} - s_t^*$ are Z-uniform for any $1 \le j \le N$ at any scale $\rho \ge 1 - 1/(80\log(1/Z))$ when $Z = o((NT)^{-2})$. Moreover, the loss of the algorithm with respect to the base strategy is at most $2ZN\sqrt{T}$.*

The proof of Theorem 11 is given in the full version.

References

[1] J.-Y. Audibert and S. Bubeck. Minimax policies for adversarial and stochastic bandits. *COLT*, 2009.

[2] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. Schapire. The nonstochastic multi-armed bandit problem. *SIAM J. Comput.*, 32:48–77, 2002.

[3] A. Blum and Y. Mansour. From external to internal regret. *Journal of Machine Learning Research*, pages 1307–1324, 2007.

[4] K. Chaudhuri, Y. Freund, and D. Hsu. A parameter free hedging algorithm. *NIPS*, 2009.

[5] T. Cover. Behaviour of sequential predictors of binary sequences. *Transactions of the Fourth Prague Conference on Information Theory, Statistical Decision Functions, Random Processes*, 1965.

[6] T. Cover. Universal portfolios. *Mathematical Finance*, 1991.

[7] E. Even-Dar, M. Kearns, Y. Mansour, and J. Wortman. Regret to the best vs. regret to the average. *Machine Learning*, 72:21–37, 2008.

[8] Y. Freund. Predicting a binary sequence almost as well as the optimal biased coin. *COLT*, 1996.

[9] Y. Freund, R. E. Schapire, Y. Singer, and M. K. Warmuth. Using and combining predictors that specialize. *STOC*, pages 334–343, 1997.

[10] E. Hazan and C. Seshadhri. Efficient learning algorithms for changing environments (full version available at http://eccc.hpi-web.de/eccc-reports/2007/tr07-088/index.html). *ICML*, pages 393–400, 2009.

[11] M. Kapralov and R. Panigrahy. Prediction without loss in multi-armed bandit problems. *http://arxiv.org/abs/1008.3672*, 2010.

[12] N. Littlestone and M.K. Warmuth. The weighted majority algorithm. *FOCS*, 1989.

[13] V. Vovk. A game of prediction with expert advice. *Journal of Computer and System Sciences*, 1998.

[14] V. Vovk. Derandomizing stochastic prediction strategies. *Machine Learning*, pages 247–282, 1999.